Java script Day1 Task

1)What will be the output of this code?

```
console.log(x);
var x=5;
O/p: undefined
```

console.log(x); is called before the variable x is assigned a value. Due to JavaScript's hoisting mechanism, the declaration (var x;) is hoisted to the top, but the assignment (x = 5;) is not. Therefore, x is declared but not yet initialized, resulting in undefined.

2)What will be the output of this code?

```
console.log(a);
var a;
O/p: undefined
```

When you call console.log(a); the variable a has been declared (due to hoisting) but not initialized with a value. Therefore, a is undefined at the time of the log statement.

3)What will be the output of this code?

```
console.log(b);
b=10;
var b;
O/p: undefined
```

When you call console.log(b);, the variable b has been declared (due to hoisting) but not yet assigned a value. So, at this point, b is undefined.

4)What will happen here?

```
console.log(c);
O/p: c is not defined
```

Variable Declaration: In JavaScript, variables must be declared before they are used. The code console.log(c); attempts to log the value of c without declaring it first.

5)What will be the output of this code?

```
console.log(e);
var e=10;
console.log(e);
e=20;
console.log(e);
O/p: undefined
    10
    20
```

First console.log(e);: At this point, the variable e is declared but not initialized. Due to hoisting, the declaration var e; is moved to the top of the scope, but the assignment (e = 10;) is not. So, console.log(e); outputs undefined.

Second console.log(e); :-After the first log, the line var e = 10; is executed, which initializes e with the value 10. Now, when you log e, it outputs 10.

Third console.log(e);:-The line e = 20; updates the value of e to 20. When you log e again, it outputs 20.

6)What will be the output oof this code?

```
console.log(f);
var f=100;
var f;
console.log(f);
O/p: undefined
    100
```

the first output is undefined because of hoisting, and the second output is 100 due to the assignment that follows. The declaration var f; is hoisted to the top but doesn't affect the output since it doesn't change the value of f.

7)What will be the output of this code?

```
console.log(g);
var g=g+1;
console.log(g);
O/p: undefined
    NaN
```

the first output is undefined due to hoisting, and the second output is NaN due to the attempt to perform an arithmetic operation with undefined.

8)What will be the output of this code?

```
var h;
console.log(h);
h=50;
console.log(h);
O/p: undefined
    50
```

The first output is undefined due to the declaration without initialization. The second output is 50 because of the assignment made before the second log.

9)What will be the output of this code?

```
console.log(i);
i=10;
```

```
var i=5;
console.log(i);
O/p: undefined
     5
```

The first output is undefined because i is declared but not yet initialized.

The second output is 5 because that’s the final value assigned to i.